

Task description - Chika wants to Cheat

Chika has a deck of N playing cards, numbered from 1 to N . She wants to play various games with her friends from the student council using these cards, but she also wants to win, so she decides to mark the cards in her deck.

The cards are all square-shaped and of size $D * D$ inches. Chika will draw a certain pattern on the back of each card, so that she will always know, by looking at the pattern, which number is written on the front of the card. She will draw the patterns using the following procedure: as many times as she wants (possibly 0 times), she picks two points that have integer coordinates relative to the bottom-left corner of the card and draws a straight line segment between them. Different methods of drawing can lead to the same pattern (for example, drawing a segment from point $(0, 0)$ to $(1, 1)$ and one from $(1, 1)$ to $(2, 2)$ leads to the same pattern as drawing a single segment from $(0, 0)$ to $(2, 2)$); Chika does not care how the pattern was built, just how it looks after it is finished.

Chika also wants to make sure that she will recognize her cards regardless of how they are rotated. A card can be rotated 0, 90, 180 or 270 degrees counterclockwise with respect to the original orientation.

Your task is to help Chika design the patterns for Q cards in her deck.

Implementation

This is a multi-run problem, meaning that you will need to implement two functions: one will need to return the pattern that goes on the back of a given card, and the other will need to return the number on the front of the card that has a given pattern. These functions will be called in separate runs of your program.

The first function,

```
vector<pair<pair<int, int>, pair<int, int>>> Code(string N, int D);
```

takes as parameters N , the number, represented as a string, that is written on the front of the card, and D , the size of the card. This function needs to return a vector containing the segments that Chika needs to draw on the back of the card with number N in order to build the pattern. A segment is represented as a pair of points, and a point is represented as a pair (x, y) of integer coordinates relative to the bottom-left corner of the card, where $0 \leq x, y \leq D$.

This function will be called Q times by the grader in the first run of your code.

The second function,

```
string Decode(vector<pair<pair<int, int>, pair<int, int>>> P, int D);
```

takes as parameters P , a vector of segments describing the pattern that is drawn on the back of the card, and D , the size of the playing card. This function needs to return the number N written on the front of the card for which Chika drew a pattern P' that, when rotated by either 0, 90, 180 or 270 degrees, exactly matches pattern P . It is guaranteed that such a pattern P' was returned as output on a previous call to function `Code`.

This function will be called Q times by the grader in a second run of your code. Each call will match a previous call to the `Code` function in the first run of your code.

Constraints

- It is guaranteed that distinct patterns for all numbers from 1 to N can be constructed for a given card size D .
- $1 \leq N$
- $1 \leq Q \leq 1000$

In addition to these constraints, the following also hold for each subtask:

- (5 points): $D = 4$, $N \leq 200$
- (10 points): $D = 2$, $N \leq 1.6 \cdot 10^7$
- (10 points): $D = 2$, $N \leq 4 \cdot 10^7$
- (40 points): $D = 2$, $N \leq 6.7 \cdot 10^7$
- (15 points): $D = 4$, $N \leq 10^{59}$
- (10 points): $D = 4$, $N \leq 2 \cdot 10^{59}$
- (10 points): $D = 4$, $N \leq 4 \cdot 10^{59}$

Sample interaction

Grader Call	Contestant Return Value	Explanation
First run begins.	-	-
<code>Code("3", 2)</code>	<pre>{{{0, 0}, {1, 1}} {{1, 1}, {0, 2}}}</pre>	<p>We have to create a pattern for number 3 on a 2×2 sized card.</p> <p>We draw 2 segments:</p> <ul style="list-style-type: none">• (0, 0) to (1, 1)• (1, 1) to (0, 2)

<code>Code("1", 2)</code>	<code>{{{0, 0}, {2, 2}}}</code>	<p>We have to create a pattern for number 1 on a 2 * 2 sized card.</p> <p>We draw 1 segment:</p> <ul style="list-style-type: none"> • (0, 0) to (2, 2)
First run ends.	-	-
Second run begins.	-	-
<code>Decode(</code> <code>{{{0, 0}, {1, 1}},</code> <code>{{1, 1}, {2, 2}}},</code> <code>2)</code>	"1"	<p>We get a pattern made up of two segments:</p> <ul style="list-style-type: none"> • (0, 0) to (1, 1) • (1, 1) to (2, 2) <p>on a 2 * 2 sized card.</p> <p>This is the same pattern as we would get from drawing the segment:</p> <ul style="list-style-type: none"> • (0, 0) to (2, 2) <p>This is exactly the same pattern with the same orientation we produced for the second call to function <code>Code</code>.</p> <p>Therefore, we return <code>N = 1</code> as a string.</p>
<code>Decode(</code> <code>{{{2, 2}, {1, 1}}</code> <code>{{2, 0}, {1, 1}}},</code> <code>2)</code>	"3"	<p>We get a pattern made up of two segments:</p> <ul style="list-style-type: none"> • (2, 2) to (1, 1) • (2, 0) to (1, 1) <p>on a 2 * 2 sized card.</p> <p>This is the pattern we received on the first call to the <code>Code</code> function, rotated by 180 degrees.</p> <p>Therefore, we return <code>N = 3</code> as a string.</p>
Second run ends.		